

AMENDMENTS TO THE SPECIFICATION

Please amend the text at Page 2, lines 14-23 and Page 3, lines 1-13 of the specification as follows:

This application is related to U.S. patent application serial number
[[_____]09/872,235, entitled “Techniques For Automatically
Provisioning A Database Over A Wide Area Network,” Attorney Docket Number 50277-1511,
filed together herewith, invented by Jean-Louis Baffier, Marc Dangeard, Thierry Gruet, Olaf
Ziegler, Dean Tan, Christina DeMello, Pascal Sero, and Kevin Walsh.

This application is related to U.S. patent application serial number
[[_____]09/872,234, entitled “Techniques For Automatically
Installing And Configuring Database Applications,” Attorney Docket Number 50277-1512, filed
together herewith, invented by Dean Tan, Marc Dangeard, Pascal Sero, and Kevin Walsh.

This application is related to U.S. patent application serial number
[[_____]09/872,067, entitled “Techniques For Providing Off-Host
Storage For A Database Application,” Attorney Docket Number 50277-1513, filed together
herewith, invented by Niloy Banerjee, John John E. So, and Michael Rocha.

This application is related to U.S. patent application serial number
[[_____]09/872,962, entitled “Techniques For Automatically
Discovering A Database Device On A Network,” Attorney Docket Number 50277-1514, filed
together herewith[[_____]], invented by Yuanjiang Ou, Sivakumar Thivakaram,
Darryl Presley and Venkat Malla.

This application is related to U.S. patent application serial number
[[_____]09/873,042, entitled “Techniques For Managing Database
Systems with a Community Server,” Attorney Docket Number 50277-1515, filed together
herewith, invented by Peter Renner, John-Brown Evans, Christina Demello, Pascal Sero, and
Kevin Walsh.

Please amend the paragraph beginning at Page 5, line 3 of the specification as follows:

A2
One or more programmers are required to produce a resource that interacts with a user's browser to (1) present information, (2) receive input, (3) make secure transactions, and (4) access a database. The presentation of textual, graphical and audio information requires a programmer skilled in the use of a markup language such as the Hyper-Text Markup Language (HTML) or the more flexible Extensible Markup Language (XML). To receive user input in response to the presented information requires a programmer able to create scripts of procedures accessed via Common Gateway Interface (CGI), or applications written in a language like JAVA (a language that runs on a software platform) that can be launched from a user's Web browser. To include information in the resource that is retrieved from a database, either initially or in response to the user input, requires a database programmer as well.

Please amend the paragraph beginning at Page 18, line 21 of the specification as follows:

A3
As used herein, the term database appliance includes any database device with a special purpose operating system tailored to the database process running on the device, including the database server appliance 220 and the database storage appliance 230 described next. The main distinction among database appliances is the function provided by the database process running on the appliance. Other examples of database appliances include an internet appliance which supports a connection with the Internet, a database Web server appliance (for example, an ORACLE 8i appliance (a web server appliance sold by ORACLE Corporation) which provides access to a database system through a user's Web browser, and a database web site building appliance which provides a wizard for building a series of Web pages with database components and is described in greater detail later below.

Please amend the paragraph beginning at Page 25, line 10 of the specification as follows:

A4
The IDSP platform 332 performs as a database device manager. It performs many database device management functions automatically. When action by the service provider 302 is required, an agent of the service provider 302, such as a database administrator employed by the service provider, may operate the IDSP system 330 through the IDSP console 334. The IDSP console runs a process, such as the ORACLE Enterprise Manager (OEM) (a manager for database and other software components), for allowing a human database administrator to interact with the IDSP platform and its processes. Alternatively, the process running on the console can be as simple as a Web browser with most of the screens presented to the administrator being generated on the IDSP platform.

Please amend the paragraph beginning at Page 35, line 1 of the specification as follows:

A5
After the customer has input information into the input items and chosen to submit the information to the registration process, a customer's inputs are validated in step 530. This validation may be performed on the IDSP platform itself, or may be performed on the customer's client process using routines downloaded to the customer's client process in a manner known in the art, such as with a Common Gateway Interface (CGI) script or a JAVA applet (an applet written in JAVA – a programming language). The validation includes checks on the customer inputs, such as ensuring that every required field is filled. In step 535 the result of the validation is checked. If validation failed, control returns to step 520. If the customer inputs were found to be valid, control passes to step 540.

Please amend the paragraph beginning at Page 36, line 9 of the specification as follows:

A6
FIG. 5B is a diagram of components on a screen presented to a customer for receiving customer profile input during the self-registration process according to one embodiment of the disclosed techniques. This customer profile screen 522 includes a list of labels 523 indicating information to be input by the customer using the screen. Adjacent to each label 523 is input area 524 where a customer may enter alphabetical or numeric characters that represent the

ALP requested information. This screen can be customized by the service provider, for example, with a logo 525 of the service provider. Customization of presentations used throughout the self-service database provisioning process can be performed as a database service selectable by the database administrator while operating the IDSP provisioning process, described later. When the customer is finished putting the information requested into the customer profile screen 522, the customer activates the submit button 521. This screen 522 is an example of forms that can be presented on a customer's platform by a client process, such as a browser executing a CGI script or JAVA applet (an applet written in JAVA a programming language).

Please amend the paragraph beginning at Page 46, line 18 of the specification as follows:

AM When the customer has completed the wizard process, the wizard uses the customer site XML file to automatically generate the objects and code required to implement the customer site. According to one embodiment, statements within the customer site XML file are used with a translator to convert the statements to commands understood by a database server, as in step 660, described above. The database server responds to the commands by building a database to support those components. A web site is then built with the web site builder based on that database, as in step 670, described above. In one embodiment, the commercial product ~~WebDB~~ WEBDB (software to enable a database supported web site) of the ORACLE™ Corporation is used to both store components and pages in a database and to dynamically generate a page of the web site based on information in the database in response to a visitor request for that page. More detail on how this is accomplished in an example embodiment is described in a later section.

Please amend the paragraph beginning at Page 61, line 25 of the specification as follows:

AS In line 8, the URL element is included; and the open and close tags, "<url>" and "</url>," indicate that the URL of the procedure that performs the service is "dca.dca_forum_main.forumQueryPage." In this embodiment, a URL of this form indicates an ORACLE™ PL/SQL (a database language) procedure "forumQueryPage" in a package named

A8
“dca_forum_main” in a package named “dca” in the same directory as the document having the XML code of Table 2. PL/SQL is an extension to structured query language (SQL) that is proprietary to the ORACLE™ Corporation. In the PL/SQL system a package is a collection of procedures that can be called externally; thus the package provides an application program interface (API). In other embodiments the procedure associated with a service can reside on another location on the network. In yet other embodiments, the procedure is not within a PL/SQL package but is any program that can be launched in response to input from a user at a browser. Such programs include a JavaScript-JAVAScript (a scripting language) that is downloaded by a browser and executed on the client’s machine, and a server side operating system script, and a JAVA servlet (servelet written in JAVA – a programming language) of a web server process, and a stand alone server side application.

Please amend the paragraph beginning at Page 71, line 13 of the specification as follows:

A9
In lines 29 through 43 the first item element is included in the folders element corresponding to the home page. The first item element involves the open and close tags, “<item>” on line 29 and “</item>” on line 43, respectively, and the intervening elements. This first item element indicates that a news component is intended for the home page by virtue of the “news” text contained in the name element on line 31. This component is customizable by the web site designer as indicated by the “CUSTOM” content of the type element in line 30. In line 32 of Table 3, the item category element is included; and the open and close tags, “<category>” and “</category>,” indicate the category is “REGULAR.” This category element is used to group components when they are inserted into the web page by the web site building routine, such as by WebDBWEBDB (software to enable a database supported web site). By placing this component in a “REGULAR” category it is treated the same way as other components in the that category. This category may be used to distinguish components that are supported by procedures that execute on a DotCom appliance from components that are supported by procedures that execute on another or remote platform. In line 33 of Table 3, the item display element is

A9
included; and the open and close tags, "<display>" and "</display>," indicate the display of the news component is "INPLACE," which means that the web page displays the actual component rather than including simply a link to a resource that displays the component if the link is selected. In line 34 of Table 3, the item output-folder element is included; and the open and close tags, "<output-folder>" and "</output-folder>," enclose no data identifying an output folder. The output-folder element may be modified by the web site designer to include a name of a directory and a file where any output generated by the component, such as a message pulled from a discussion group, may be stored temporarily. This is especially useful when the component is provided by an external resource such as a remote database that places results from a query in a file on the remote system.

Please amend the paragraph beginning at Page 82, line 7 of the specification as follows:

A10
When all changes are made the user selects the submit button 696 (labeled with the value "OK"). Upon clicking the submit button, following the HTML code inserted on the form by line 82 of the XSLT statements in Table 4, the designer's browser calls the "formSubmit()" method. This method is found in the ~~Javascript~~ JAVAScript (a scripting language) inserted into this HTML page by the XSLT statements in lines 8 through 14 of Table 4. This script obtains values for the site identification and occurrence of the present form and loads those values into the HTML hidden variables (p_site_id and p_occurrence) then calls the HTML submit method associated with this form. The HTML submit method was specified in the HTML statements inserted into the HTML page by the XSLT statements in lines 16, 17, 19 and 20. The designer's browser passes the designer's input along with the values of the p_site_id and p_occurrence hidden variables to the procedure "dca.dca_forum_data.set_args" using the POST method. In this embodiment, the procedure set_args is a method in an object called "dca_forum_data" residing in the package called "dca." As explained above, in this embodiment, the dca package is a PL/SQL package. The set_args procedure creates an XML element called "item" for the customer site XML file following the DTD for a site-template document. For example, if the

ALL
designer typed "ALL" into input text box 693 then pressed the OK button 696, the set_args procedure would generate the XML statements constituting the arguments element found in Table 3 in lines 50 through 55.

Please amend the paragraph beginning at Page 87, line 17 of the specification as follows:

ALL
The database wizard comes with templates already installed. As described above in step 687, the appliance owner can customize individual components and edit the templates to generate new templates. According to one embodiment, a template, for example as represented by an XML file, is derived from an existing web site. The web site could have been built by a programmer operating the database web site builder without benefit of a wizard. Using this embodiment, an XML file describing a template is derived from the Web page presented to the visitor's browser. For example, if a web site has been produced by WebDB WEBDB (software to enable a database supported web site) using a PL/SQL procedure call that creates a dynamic HTML page which invokes a Java JAVA applet (an applet written in JAVA – a programming language), then the parameters of the Java JAVA applet (an applet written in JAVA – a programming language) can be obtained from the PL/SQL procedure. Those parameters can be retrieved by the wizard and made available for customization by the user operating the wizard. Components that refer to resources hosted on a remote machine, for example, are represented by a URL address. In some circumstances, a component from an existing web site does not have an XML document describing its parameters, so that such a component cannot be customized.

Please amend the paragraph beginning at Page 88, line 13 of the specification as follows:

ALL
The customer site XML file is used to generate the web site. In the preferred embodiment, the HTML statements that render each component and procedures that provide functionality for the component are stored in a WebDB WEBDB (software to enable a database supported web site) database and implemented by WebDB WEBDB (software to enable a database supported web site) upon requests from a visitor's browser for a web page at the web

site. In one embodiment, some procedures are executed by a DotCom appliance or other remote device outside ~~WebDB~~ WEBDB (software to enable a database supported web site). In this case, the ~~WebDB~~ WEBDB (software to enable a database supported web site) database stores a reference to the procedure that resides on the DotCom appliance or the remote device.

AI2
For example, if the XML statements in Table 3 represent the customer site XML file after modifications, then a web site is built using ~~WebDB~~ WEBDB (software to enable a database supported web site) based on the information in Table 3 and in Table 2. ~~WebDB~~ WEBDB (software to enable a database supported web site) stores information for forming web pages in a database and constructs the pages in response to receiving a request from a visitor's browser for the page, identified by its URL.

Please amend the paragraph beginning at Page 89, line 1 of the specification as follows:

AI3
~~WebDB~~ WEBDB (software to enable a database supported web site) expects a web site designer to upload content and procedures using an hypertext transfer protocol (HTTP), that is ~~WebDB~~ WEBDB (software to enable a database supported web site) expects requests as a URL with parameters attached, and ~~WebDB~~ WEBDB (software to enable a database supported web site) responds to those requests with HTML pages. ~~WebDB~~ WEBDB (software to enable a database supported web site) organizes web site content into a series of folders that represent pages and items that represent components. However, ~~WebDB~~ WEBDB (software to enable a database supported web site) does not provide the full set of components provided by the wizard nor templates of folders associated with a category of activities as provided by the wizard.

In one embodiment, the wizard sends HTTP requests to a ~~WebDB~~ WEBDB (software to enable a database supported web site) server, intercepts the responses and, without presenting the ~~WebDB~~ WEBDB (software to enable a database supported web site) HTML documents to the designer, inputs text into the forms and follows the submit methods automatically, based on the information in the XML documents.

AI3
In the preferred embodiment, the web site building wizard uses the PL/SQL procedures that the ~~WebDB~~-WEBDB (software to enable a database supported web site) server uses to populate the database that supports the web pages of a web site. The wizard uses the ~~WebDB~~-WEBDB (software to enable a database supported web site) PL/SQL routines to open an account on the database server for the web site designer, create database objects such as tables, indexes, views, synonyms, and stored procedures for storing and retrieving the properties of the folders, items and components in the XML documents. The wizard then uses the PL/SQL routines to load the information from the XML documents into those tables, including the name of procedures for providing component functionality and the source of the arguments to pass when invoking the procedure.

For example, the wizard uses the ~~WebDB~~-WEBDB (software to enable a database supported web site) PL/SQL routines to create a components table having fields that include name, title, run service, and admin service and an arguments table having fields that include a component name field, an argument position field and an argument value field. The wizard then loads values into the components table into a row corresponding to a particular discussionforum component based on information in the XML documents provided to the wizard by the site designer. The row in the components table after loading is shown in Table 5.

Please amend the paragraph beginning at Page 90, line 9 of the specification as follows:

AI4
After the folders and items are loaded, a visitor may request a folder by its URL; and ~~WebDB~~-WEBDB (software to enable a database supported web site) assembles the components into an HTML document, launching any procedures required to generate the component's HTML with the associated argument values (whether stored or based on visitor input). Child folders are included as links. Image files are retrieved based on their URL address rather than being stored in the ~~WebDB~~-WEBDB (software to enable a database supported web site) database. The procedure launched may be a stored procedure within the database or may be a remote procedure which has its URL stored in the ~~WebDB~~-WEBDB (software to enable a

AI4
database supported web site) database. If this reference is to a stored procedure, then the stored procedure would be run with the parameters listed in the arguments table. A stored procedure may be a PL/SQL procedure that creates or uses a database object, such as a procedure to query a database of products in the inventory of a retailer. If this is a reference to a remote resource, then a request to the URL of the remote resource would be made with the associated arguments in Table 6.

Please amend the paragraph beginning at Page 91, line 1 of the specification as follows:

For example, if a visitor to the home page selects a link to the discussionforum page, then ~~WebDB-WEBDB~~ (software to enable a database supported web site) will construct the HTML document to send as a response based on the components on the page. To construct the discussionforum component on the discussionforum page, the component is found in Table 5 and the component run service is invoked. This service references a PL/SQL package that resides on the DotCom appliance. In this case ~~WebDB-WEBDB~~ (software to enable a database supported web site) sends the arguments in Table 6 associated with this particular component to the DotCom appliance. The called procedure performs the functions and generates the result, if any, in an HTML document for display by the visitor's browser.

AI5
~~WebDB-WEBDB~~ (software to enable a database supported web site) determines whether the visitor has the privileges required to perform the requested action. For example, if the visitor requesting the discussionforum component has administrator privileges, then ~~WebDB-WEBDB~~ (software to enable a database supported web site) launches the procedure stored in the admin service field of the components table rather than the procedure in the run service field. If not, then the procedure in the run service field is launched. In this way different visitors may access different capabilities. This process allows the site administrator to perform special functions after the site becomes operational. The extra functions offered by the admin service may include changing the users allowed to access the discussionforum, changing the configuration of the discussionforum, editing messages, entering a message, uploading a stored message and

A15
downloading a message to storage. These functions are offered to the administrator in an HTML form not available to less privileged visitors to the web site.

Please amend the paragraph beginning at Page 105, line 23 of the specification as follows:

A16
Process 1320 is used to monitor appliances and other database devices through the display on the IDSP console (item 334 in FIG. 3A). In this embodiment the monitoring is done using a Java JAVA applet (an applet written in JAVA – a programming language) that runs on the console. This is an advantage because the same Java JAVA applet (an applet written in JAVA – a programming language) can be run from a browser, and that allows the platform owner to monitor the appliances over the Internet, or other network, when the platform owner is separated from the IDSP site. Although a Java JAVA applet (an applet written in JAVA – a programming language) is used in this embodiment, any script or process known in the art can be used for communicating between a remote terminal and the IDSP platform or database devices. The console will have a form-based interface as well as a Web-based interface, so that all operations can be handled remotely. It is expected that the console will be adapted from an enterprise manager appliance pack available for the database appliances.
